

**A report on**

**“APPLE LEAF DISEASE DETECTION”**

**Submitted in partial fulfilment for the award of the degree of**

**BACHELOR OF TECHNOLOGY (HONOURS)**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**(DATA SCIENCE)**

**Submitted by**

**AYUSH JAIN**

**18BTRCR006**

**BHUPENDRA GUPTA**

**18BTRCR007**

**DEEKSHA YADAV**

**18BTRCR011**

**KAVEEN GANDHI**

**18BTRCR019**

**Under the guidance of**

**Prof. Mohammed Zabeeulla**

**DESIGNATION**

**Faculty of Engineering & Technology**

**Jain (Deemed-To-Be University)**

**Department of Computer Science & Engineering**

Jain Global Campus, Kanakapura Taluk - 562112

Ramanagara District, Karnataka, India

2020-2021



**Department of Computer Science & Engineering**

Jain Global Campus, Kanakapura Taluk - 562112

Ramanagara District, Karnataka, India

# **CERTIFICATE**

This is to certify that the project work titled **“APPLE LEAF DISEASE DETECTION”** is carried out by **STUDENT NAME (USN NUMBER)**, a bonafide students of Bachelor of Technology at the Faculty of Engineering & Technology, Jain (Deemed-to-be University), Bangalore in partial fulfilment for the award of degree, Bachelor of Technology (Honours) in Computer Science & Engineering (Data Science), during the Academic year **2020-2021**.

|  |  |  |
| --- | --- | --- |
| **Guide name** **Guide** Designation  Dept. of CSE,  Faculty of Engineering & Technology,  Jain (Deemed-to-be University)  Date: | **Dr. Kuldeep Sharma**  Head of the Department,  Dept. of CSE,  Faculty of Engineering & Technology,  Jain (Deemed-to-be University)  Date: | **Dr. Hariprasad S A**  Director,  Faculty of Engineering & Technology,  Jain (Deemed- to-be University)  Date: |

Name of the Examiner Signature of Examiner

1.

2.

# **DECLARATION**

We, **AYUSH JAIN(18BTRCR006), BHUPENDRA GUPTA(18BTRCR007), DEEKSHA YADAV(18BTRCR011), KAVEEN GANDHI(18BTRCR019)**, are students of sixth semester B. Tech (Honours) in **Computer Science & Engineering (Data Science)**, at Faculty of Engineering & Technology, **Jain (Deemed-To-Be University)**, hereby declare that the project work titled **“APPLE LEAF DISEASE DETECTION”** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology (Honours) in** **Computer Science & Engineering (Data Science)** during the academic year **2020-2021**. Further, the matter presented in the project has not been submitted previously by anybody for the award of any degree or any diploma to any other University, to the best of our knowledge and faith.

Signature

AYUSH JAIN

USN: 18BTRCR006

BHUPENDRA GUPTA

USN: 18BTRCR007

DEEKSHA YADAV

USN: 18BTRCR011

KAVEEN GANDHI

USN: 18BTRCR019

Place: Bangalore

Date: 15- 06 -2021

# **ACKNOWLEDGEMENT**

*It is a great pleasure for us to acknowledge the assistance and support of a large number of individuals who have been responsible for the successful completion of this project work.*

*First, we take this opportunity to express our sincere gratitude to* ***Faculty of Engineering & Technology, Jain (Deemed-to-be University),*** *for providing us with a great opportunity to pursue our Bachelor’s Degree (Honours) in this institution.*

*In particular we would like to thank* ***Dr. Hariprasad S A****,* ***Director****,* ***Faculty of Engineering & Technology****,* ***Jain (Deemed-to-be University),*** *for his constant encouragement and expert advice.*

*It is a matter of immense pleasure to express our sincere thanks to* ***Dr. Kuldeep Sharma****,* ***Head****,* ***Department of Computer Science & Engineering****,* ***Jain (Deemed-to-be University),*** *for providing right academic guidance that made our task possible.*

*It is a matter of immense pleasure to express our sincere thanks to* ***Dr. D Stalin Alex****,* ***Program Incharge, Prof. Mohammed Zabeeulla, Program Coordinator of the department****,* ***Computer Science & Engineering (Data Science)****,* ***Jain (Deemed-to-be University),*** *for providing right academic guidance that made our task possible.*

*We would like to thank our guide* ***Guide name, designation****,* ***Dept. of Computer Science & Engineering****,* ***Jain (Deemed-to-be University),*** *for sparing his valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

*We would like to thank our Project Coordinator* ***Prof. Neha Gautam*** *and all the staff members of Computer Science & Engineering for their support.*

*We are also grateful to our family and friends who provided us with every requirement throughout the course.*

*We would like to thank one and all who directly or indirectly helped us in completing the Project work successfully.*

*Signature of Students*

# **TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CERTIFICATE** | |  |
| **DECLARATION** | |  |
| **ACKNOWLEDGEMENT** | |  |
| **TABLE OF CONTENT** | |  |
| **ABSTRACT** | |  |
| **LIST OF FIGURES** | |  |
| **LIST OF TABLES** | |  |
| **NOMENCLATURE USED** | |  |
| **Chapter 1** | |  |
| 1. **INTRODUCTION** | |  |
|  | * 1. Overview |  |
|  | * 1. Problem Definition |  |
|  | * 1. Objectives |  |
|  | * 1. Methodology |  |
|  | * 1. Hardware and Software Tools Used |  |
| **Chapter 2** | |  |
| 1. **LITERATURE SURVEY** | |  |
|  | * 1. Related Work |  |
|  | * 1. Existing System |  |
|  | * 1. Limitation of Existing System |  |
|  | * 1. Proposed System |  |
| **Chapter 3** | |  |
| 1. **METHODOLOGY** | |  |
|  | * 1. Dataset |  |
|  | * 1. Architecture |  |
|  | * 1. Sequence Diagram |  |
|  | |  |
|  | |  |
| **Chapter 4** | |  |
| 1. **IMPLEMENTATION** | |  |
| **Chapter 5** | |  |
| 1. **RESULTS AND ANALYSIS** | |  |
|  | * 1. Result Discussion |  |
|  | * 1. Comparison with Previous Studies |  |
|  | * 1. Analysis |  |
| **Chapter 6** | |  |
| 1. **CONCLUSIONS AND FUTURE SCOPE** | |  |
| **REFERENCES** | |  |
| **APPENDIX** | |  |

# **ABSTRACT**

Agriculture is the feeding source to every human being on this planet. And Indian economy highly depends on agriculture resources. So, productivity of agriculture resources is must. Due to certain factors the productivity decreases and it can be controlled by detecting the leaf disease at the right time.

Apple leaf disease is the foremost factor that restricts apple yield and quality. Usually, much time is taken for disease detection with the existing diagnostic techniques; therefore, farmers frequently miss the best time for preventing and treating diseases. The detection of apple leaf diseases is a significant research problem, and its main aim is to discover an efficient technique for disease leaf image diagnosis. This article has made an effort to propose a method that can detect the disease of apple plant leaf using deep neural network (DNN). Plant diseases detection system (PDDS) architecture is designed. Speeded up robust feature (SURF) is used for feature extraction, which helps to achieve better detection and classification accuracy. Classification parameters, such as Precision, Recall, F-measure, Error, and Accuracy is computed, and a comparative analysis has been performed to depict the effectiveness of the proposed work.

# **LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Fig. No.** | **Description of Figure** |  |
|  |  |  |

3.2.1 Architecture

3.3.1 Use - Case Diagram

4.1.1 Clahe enhancement

4.1.2 Segmented image

4.1.3 Surf extraction

4.2.1 Clahe enhancement

4.2.2 Segmented image

4.2.3 Surf extraction

4.3.1 Clahe enhancement

4.3.2 Segmented image

4.3.3 Surf extraction

5.2.1 VGG-16 Training and validation accuracy

5.2.2 VGG-16 Training and validation loss

5.3.1 CNN Training and validation accuracy

5.3.2 CNN Training and validation loss

# **LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Description of the Table** |  |
| 1.5.1  1.5.2  2.1.1 | Hardware Tools  Software Tools  Related Work |  |

# **NOMENCLATURE USED**

DNN - Deep Neural Network.

PDDS - Plant Diseases Detection System.

SURF- Speeded Up and Robust Feature.

SIFT – Scale Invariant Feature Transform.

CNN – Convolution Neural Network.

% - Percentage.

|  |  |
| --- | --- |
|  |  |

# **Chapter 1**

**INTRODUCTION**

* 1. **Overview**

The detection of Apple plant leaf disease is the vital process as plants leaf consists of multiple disease regions. The shape of each disease in the leaves varies, having uneven colour and fuzzy texture. There are many different ways to detect the leaf disease through soft-computing based approaches, image pre-processing, automatic detection. The general diseases are in the form of spots, which are yellow and brown. Alternatively, Segmentation based approaches helps in recognising the leaf disease and it separates the leaf from the background. The different types of image processing techniques are applied to process those images, to get different and useful features needed for the purpose of analyzing the disease.

**1.2. Problem Definition**

To detect the different types of Apple Leaf Disease using Deep Learning Techniques and Algorithms.

**1.3. Objectives**

1. Classification of different types of Disease in Apple plant.

**1.4. Methodology**

The step-by-step approach for the proposed Leaf disease detection:

1. Acquiring Plant Leaf Disease dataset is the very first step.

2. Image Enhancement is applied to the uploaded plant leaf images. Image enhancement technique is used to help in the segmentation process of the image which split the leaf image into background and foreground

3) In this process, the K-means algorithm is applied using morphological operators. K-means technique is used to differentiate foreground and background of leaf image using 2 clusters. As the required leaf image is foreground image; therefore, by applying K-means algorithm foreground image has been extracted.

(4) To achieve only the desired features from the whole leaf image, we will apply SURF as a feature extraction technique.

(5) Based on the mentioned algorithms and methodology steps, the features and labels will be given to the CNN model to classify the diseases.

* 1. **Hardware and Software Tools Used**

**Hardware:**

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Type1 | Type2 | Type3 |
| Laptop | Lenovo | MacBook | HP |
| OS | Windows | Mac OS | Windows |
| Gpu | 2gb | None | 4gb |
| Ram | 8gb | 4gb | 8gb |
| SSD | 1TB | 500gb | 2TB |
| Processor | Intel core i5 | Intel core i3 | Intel Core i5 |

***Table 1.5.1 – Hardware tools***

**Software:**

|  |  |
| --- | --- |
| Item | Name |
| App | Anaconda Navigator |
| Open Source | Jupyter Notebook |
| Language | Python3 |
| Environment | Base(root), Tensorflow |
| IDE | Python |
| Main Modules | Opencv, keras ,tensorflow |

***Table 1.5.2 – software tools***

# **Chapter 2**

**LITERATURE SURVEY**

* 1. **Related Work**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.no** | **Publisher Name** | **Year** | **Dataset** | **Algorithms used** | **Accuracy** | **Limitations** |
| 1. | Trimi Neha tete, Sushma kamlu | 2017 | Plant Village | K-means clustering and Neural Networks (NN) | **71.7%** | Multiple diseases may occur on the same leaf. |
| 2. | Zhang Chuanlei, Zhang Shanwen, Yang Jucheng, Shi Yancui | 2017 | Apple leaf disease dataset | RGA, GA-CFS and SVM | **93%** | The diseased leaf image  recognition system should be capable of processing the  diseased leaf images acquired in different ways. |
| 3. | Liya Bin, Yun Zhang, Yuxiang Li, DongJian He | 2017 | Apple leaf disease dataset | PCA jittering, NAG Algorithm, GoogLeNet Inception. | **97.62%** | Determining the best structures of the network model is fundamentally a more difﬁcult task. |
| 4. | Swati Singh and Sheifali Gupta | 2018 | Created own dataset | K-means Clustering, (CCV), (LBP), (GCH) (CLBP) and SVM. | **98%** | Multiple diseases may occur on the same leaf. |
| 5. | Saraansh Baranwal,  Siddhant Khandelwal,  Anuja Arora | 2019 | Apple leaf disease dataset | LeNet, GoogleNet , AlexNet, SVM, KNN | **98.42%** | Without dropout and hypertuning received lower results. |
| 6. | Peng Jiang, Yuehan Chen, Bin Liu | 2019 | Apple leaf disease dataset | GoogLeNet Inception module and integrating Rainbow concatenation. | **78.80%** | Multiple diseases may occur on the same leaf. |
| 7. | Marko Arsenovic, Mirjana Karanovic, Srdjan Sladojevuc | 2019 | Apple leaf disease dataset | GAN and Deep Learning Architecture | **93.67%** | The proposed methods cannot detect multiple diseases in one image or cannot detect multiple occurrences of the same diseases in one image |
| 8. | Jalal Sadoon Hameed Al-bayati, Burak Berk Ustündag | 2019 | Plant Village Dataset Master | K-means Clustering, SURF, GOA, DNN | **98.23%** | we will investigate and evaluate our methodology in other practical environments with many infected leaves of several plants. |
| 9. | Dr.L.Malathi, P.Yogashree, A.Thamaraiselvi | 2020 | Apple leaf disease dataset | DCNN | **98%** | Available datasets do not contain images gathered and labeled from real-life situations. |
| 10. | Hee-Jin Yu and Chang-Hwan Son | 2020 | Apple leaf disease dataset | FS-SubNet, LSA-Net | **89.4%** | Extracting more discriminative features and semantic-level spot information |

***Table 2.1.1 – Related work***

* 1. **Existing System**

Earlier papers are describing to detect mainly through pre-processing , Normalizing and training the model. These were the direct steps for any image based model approach. But, the evolution in deep learning helped to change the transitions of image based models approach. This models have a capacity to carry out large number of features. This type of learning is called transfer learning. The base of this models comes from CNN model. For large datasets, the transfer learning approach is used in the existing system. There are various methods to detect the apple leaf disease. Some are diagnostic, object detection and through transfer learning. Diagnostic techniques is through enhancement, segmentation, feature extraction and feature optimization. Object detection is also the best technique to use, because the location of the disease can be spotted from it. The detection through diagnostic techniques will be laborious task, because the amount time will be considerable more compared to transfer layer techniques. But diagnostic techniques will help in gaining the knowledge of pre-processing, selecting features and optimizers. Transfer learning models will do all this operations in it and give the resultant output. For classifying the disease we can use any classification algorithm and check the accuracy. These are the some of the existing systems for apple leaf disease detection.

* 1. **Limitation of Existing System**

1.The implementation still lacks in accuracy of result in some cases. More optimization is needed.

2. Priori information is needed for segmentation.

3.Database extension is needed in order to reach the more accuracy.

4.Very few diseases have been covered. So, work needs to be extended to cover more diseases.

5.The possible reasons that can lead to misclassifications can be as follows: disease symptoms varies from one plant to another, features optimization is needed, more training samples are needed in order to cover more cases and to predict the disease more accurately.

* 1. **Proposed System**

This section will help you to know the method we used to detect the disease of an Apple leaf disease detection.

The first most step is to acquire the data of Apple lead disease. You will find various datasets of Apple leaf disease ,but we preferred to take it from Kaggle. Because the dataset included the separate categories of Apple leaf disease. We chose four disease those are Apple Scab, Apple Black Rot, Apple Cedar Rust and Apple healthy. The objective is to detect the Apple disease using Segmentation and CNN model .

The detection of disease is not an easier task because the image quality should be clear ,so that while feature extraction we can pick relevant features for training. There are various techniques for enhancing an image, but we used Contrast Limited Adaptive Histogram Equalization (CLAHE), because it takes cares of over-amplification of the contrast. It operators on small regions in the image, called tiles, rather than the entire image. The neighbouring tiles are then combined using bilinear interpolation to remove artificial boundaries . This algorithm can be applied to improve the contrast of images.

The next step is segmenting the enhanced the image into two parts, one is background and other is the foreground(leaf part). The main purpose of segmentation is to separate the leaf with the background. The use of segmentation will be helped in achieving a higher accuracy for our model. For segmentation, we use K-means Clustering technique, because it is an algorithm where we can segment the image on any number of clusters. It can segment on the number of clusters provided. It calculates the distance from the centroids and divide the points into the categories. We dividing the image into background and foreground, so we gave number of clusters as two. So, the image will be segmented into two clusters. The use of clusters can be identifying when we do feature extraction.

While extracting the features from the apple leaf some features are not important in the background part. Because there can be a higher pixel value in the background of the apple leaf, which can affect the model performance. So, for those problems not to occur we used masking. Masking is the technique to mask the particular area based on requirement.

So, far I mentioned about feature extraction . Feature Extraction is the important step in training the models. Because a machine can’t directly understand the images. It understands using features. We can say that features are very important for our model. There are many features in an image, we have to extract the features which are important for training .We applied Speeded Up Robust Features(SURF) algorithm for feature extraction. It is a fast and robust algorithm for local, similarity invariant representation and comparison of images. This approach lies in in its fast computation of operators using box filters, thus enabling real-time applications. With this technique we will do feature extraction.

For training we will be using CNN model. The features will be model lo learn about the diseases and classify the disease. The objective is to detect which type of disease it is.

Finally, the model is applied on the test dataset and the Accuracy, Recall and various other parameters.

# **Chapter 3**

**METHODOLOGY**

* 1. **Dataset**

To detect and classify the plant diseases “Apple Leaf Diseases” initially data is collected from the dataset named as “Plant Village Dataset Master”. In this dataset, four diseases from which apple leaves are presented, namely, apple scab, black rot, cedar-apple rust, and healthy leaves.

* 1. **Architecture**

This is the Architecture of any Plant Leaf Disease Detection. The steps involved in creating a model for Plant Leaf Disease are as follows:

1. Input leaf
2. Pre-processing
3. Feature Extraction
4. Feature optimization
5. Training
6. Testing
7. Classification

The last step depends on the type of model you choose. Models like CNN, VGG-16 can classify themselves. If you don’t use the above mentioned models, then classification models like SVM, Decision Tree etc. can be used.

**Input Leaf Image**

**Pre-Processing**

**Image Enhancement**

**Image Segmentation**

**Feature Extraction**

**Feature optimization**

**Training using CNN**

**Trained Database**

**Test Leaf Image**

**Classification**

**Is Matched?**

**No**

**Yes**

***fig 3.2.1 - Architecture***

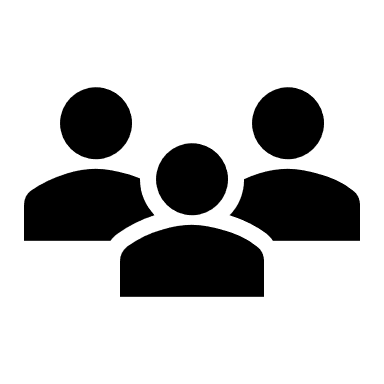
* 1. **Sequence Diagram**

****

v

USER

v

****

SERVER

***fig 3.3.1 – Use Case Diagram***

**Use-case Diagram is the architecture, it defines the process of implementation.**

**The basic terminologies remain same as below:**

1. **Leaf images.**
2. **Image Enhancement**

**CLAHE** - CLAHE is a variant of Adaptive histogram equalization (AHE) which takes care of over-amplification of the contrast. CLAHE operates on small regions in the image, called tiles, rather than the entire image. We can also apply CLAHE to colour images, where usually it is applied on the luminance channel and the results after equalizing only the luminance channel of an HSV image are much better than equalizing all the channels of the BGR image.

1. **Segmentation**

**K-means Clustering -** K-Means clustering algorithm classifies the input data points into many number of classes based on clusters inherent distances. The algorithm assigns that data features to create a vector space for clustering. These data points are clustered around centroids. where k is number of clusters Si , i = 1, 2, . . . , k and µi is the mean or centroid of all the points xj∈ Si.

1. **Feature Extraction**

**SURF -** Speeded-up robust features (**SURF**) is a scale- and rotation-invariant feature detector and descriptor used to identify regions of interest and commonly used in image recognition

1. **Model**

**Some terminologies in CNN:**

**Convolution Layer –** Convolutional layers are the layers where filters are applied to the original image, or to other feature maps in a deep CNN.

**Activation Function –** Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron.

**Batch Normalization –** Batch normalization is a technique for training very deep neural networks that standardizes the inputs to a layer for each mini-batch. This has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks.

**Max-Pooling –** Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter.

**Dropout –** The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting.

**Flatten Layer** - Flatten is the function that converts the pooled feature map to a single column that is passed to the fully connected layer.

**Dense Layer –** The dense layer is a neural network layer that is connected deeply, which means each neuron in the dense layer receives input from all neurons of its previous layer.

The model summary function will help us to summary of the model like the number of convolution layers, activation function used.

**Optimizer –** Adam optimizer is used to optimize the features.Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models**.**

1. **Performance –** Check the accuracy, analyse and compare the resultant outputs.

# **Chapter 4**

**IMPLEMENTATION**

**Steps to run:**

**Import Libraries –** Numpy, Pandas, Matplotlib, Cv2, Tensorflow, Keras, OS, Pickle, Sklearn. This are the important libraries used.

**Input Leaf-** The first-most step is to import the dataset. Due to the large size of the dataset rather than uploading the dataset to the Jupyter Notebook. You can access the dataset by providing the path to the Jupyter Notebook. Then you can read the dataset once you reach to file folder.

Dataset includes four categories:

1. Apple Scab
2. Apple Black Rot
3. Apple Cedar Rust
4. Apple Healthy

For the above four categories we will assign the labels from 0-3.

Then we uploaded the dataset into the list in the Jupyter Notebook. So that we can access all the images inside the folder.

**Pre-Processing:** The Pre-processing follows in below two steps:

**Image Enhancement –** Now, the uploaded images in a list will be given to enhance for clearly recognizing the leaf images. The enhancement technique used is Contrast Limited Adaptive Histogram Equalization (CLAHE). We will append all the enhanced images into a new list as clahe images.

**Image Segmentation –** This is the vital process. Segmentation will help in segmenting the enhanced images into background and foreground. This can be done using K-Means Clustering.

**Feature Extraction –** Feature Extraction is the extracting features from the segmented image. To extract features, we have used Speeded-up robust features (SURF). This algorithm extracts those features which are useful for training.

**Train - Test Split –** Train- Test Split is generally done in three ratio.

1. 70% Training and 30% Testing.
2. 80% Training and 20% Testing.
3. Some researchers prefer 90% Training and 10% Testing.

We prefer to take 80% Training and 20% Testing.

**Model Creation –** Before training, the model creation or to define the model structure is very important for image datasets. Because you have to decide the number of layers to use in the model. We used CNN model.

**Model Compile –** Compile the model on the loss function as categorical cross-entropy, metrics as accuracy, Optimizer as Adam.

**Model Fit –** Fit the model on the training data. With specified epochs, batch-size, learning rate and steps-per-epochs.

**Training Accuracy –** Calculatethe training accuracy.

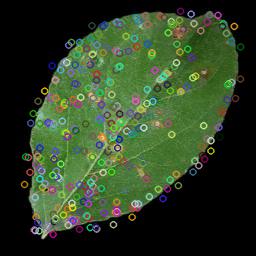
**Testing Accuracy –** Calculatethe testing accuracy.

**Precision -** Calculatethe Precision.

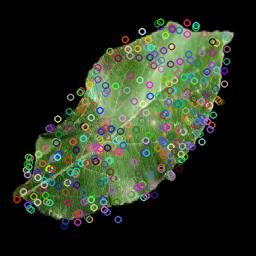
**Recall -** Calculatethe Recall.

**F-measure -** Calculatethe F-measure.

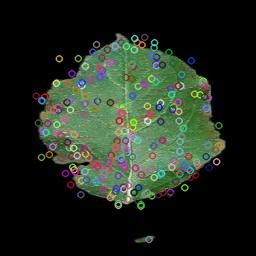
**Error –** Calculatethe error.

**  **

*Fig no: 4.1.1 Clahe enhancement,4.1.2 Segmented image, 4.1.3 Surf extraction*

**  

*Fig no: 4.2.1 Clahe enhancement, 4.2.2 Segmented image, 4.2.3 Surf extraction*

** 

*Fig no 4.3.1 Clahe enhancement, 4.3.2 Segmented image, 4.3.3 Surf extraction*

# **Chapter 5**

**RESULTS AND ANALYSIS**

* 1. **Result Discussion**

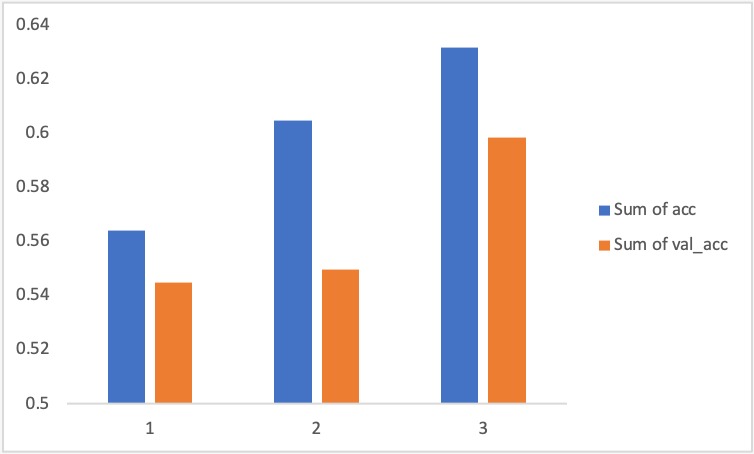
The result section will include the training and testing accuracy of our model. From this section we will know that how much our model as learned from the training and how it is performing the testing. This is the inspection of our model. For evaluation of any model, there are some parameters like training accuracy , validation accuracy and loss is computed, and a comparative analysis has been performed to depict the effectiveness of the proposed work.

* 1. **Comparison with Previous Studies**

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. The comparasion model is VGG16.

As VGG16 is a convolution neural net (CNN ) architecture which was used to win ILSVR(Imagenet) competition in 2014. It is considered to be one of the excellent vision model architecture till date. Most unique thing about VGG16 is that instead of having a large number of hyper-parameter they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a pretty large network and it has about 138 million (approx) parameters.

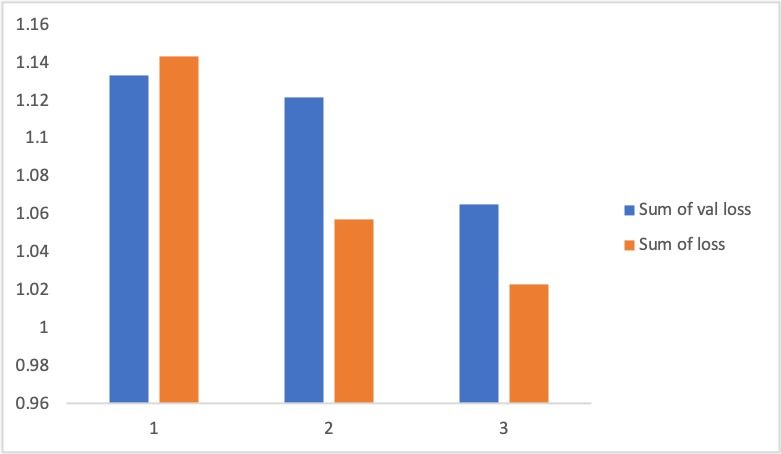
**VGG16 Training and Validation Accuracy Graph:**



*Fig 5.2.1*

This clearly indicated that accuracy is increasing as the number of epochs increases. After certain number of epochs the accuracy will be remain almost same. Even the validation accuracy is increasing with the increase in number of epochs.

**VGG16 Training and Validation Loss Graph:**

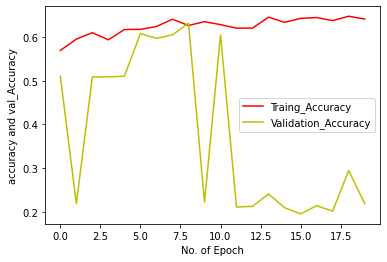


*Fig 5.2.2*

This clearly indicated that Training Loss is decreasing as the number of epochs increases. After certain number of epochs the Training Loss will be remain almost same. Even the validation Loss is decreasing with the increase in number of epochs.

* 1. **Analysis:**

**Validation Accuracy:** It is the test (or testing) **accuracy** often refers to the **validation accuracy**, that is, the **accuracy** you calculate on the data set you **do** not use for training, but you use (during the training process) for validating (or "testing") the generalisation ability of your model or for "early stopping”.

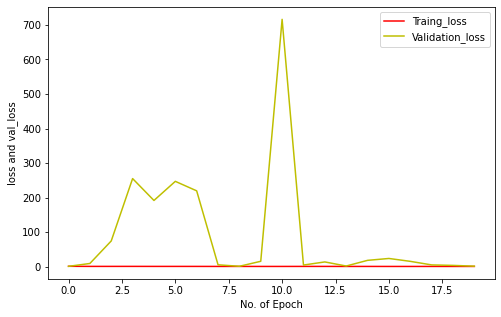
****

*Fig 5.3.1*

We achieved an Training accuracy of **0.7719** and Validation accuracy of **0.7598**.

The training accuracy can be improved by proper enhancement, segmentation with ROI on the Disease leaf part with proper feature extraction, feature optimizer. Also a proper device with high configurated like 8 to 16 GB RAM , GPU , some 4 to 8 GB graphic cards to train the model with more numbers of epochs. As the number of epochs increase our model would get train properly and in less time to achieve better accuracy.

**Validation Loss :** The **loss** is calculated on training and **validation** and its interpretation is based on how well the model is doing in these two sets. It is the sum of errors made for each example in training or **validation** sets. **Loss** value implies how poorly or well a model behaves after each iteration of optimization

****

*Fig 5.3.2*

We got a Validation Loss of **0.6609**.

To reduce the loss we need to choose a better activation function , Loss function and also Back propagation to avoid High variance and as it help in reducing the loss.

# **Chapter 7**

**CONCLUSIONS AND FUTURE SCOPE**

In this paper, experimental analysis is done to detect and classify the diseases from apple plant leaf images in an effective way using the concept of CNN as a convolutional neural network approach.

The proposed PDDS has been carried out in different steps before applying the DNN such as image enhancement and segmentation in pre-processing phase, feature extraction using SURF descriptor and at the last training and classification using the DNN. By using all these processes, the accuracy of the proposed PDDS for apple leaf has improved. From the experiments, it is concluded that the accuracy of the proposed model with SURF

In the near future, we will investigate and evaluate our methodology in other practical environments with many infected leaves of several plants and try to apply the different ways to detect the apple leaf disease.

# **REFERENCES**

* <https://annals-csis.org/proceedings/rice2017/pliks/24.pdf>
* <https://ijabe.org/index.php/ijabe/article/view/2166/pdf>
* <https://www.researchgate.net/publication/322142005_Identification_of_Apple_Leaf_Diseases_Based_on_Deep_Convolutional_Neural_Networks>
* <https://acadpubl.eu/jsi/2018-118-18/articles/18b/16.pdf>
* <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3351641>
* <https://ieeexplore.ieee.org/abstract/document/8706936>
* <https://www.mdpi.com/2073-8994/11/7/939/htm>
* <https://www.atlantis-press.com/journals/ijcis/125931940>
* <https://www.xajzkjdx.cn/gallery/67-may2020.pdf>
* <https://openaccess.thecvf.com/content_CVPRW_2020/papers/w5/Yu_Leaf_Spot_Attention_Network_for_Apple_Leaf_Disease_Identification_CVPRW_2020_paper.pdf>

# **APPENDIX**

<https://github.com/khalidmeister/apple-leaf-disease-prediction>

<https://github.com/topics/surf>

<https://github.com/topics/image-enhancement>

<https://github.com/mehra-deepak/Plant-Disease-Detection>

<https://dev.to/petercour/enhance-image-with-python-pil-222e>

<https://www.youtube.com/watch?v=6CqRnx6Ic48&t=82s>